

OpenVINO Tools для Linux

1. Установка основных компонентов набора инструмента

Откройте окно терминала и перейдите в каталог, в который был загружен файл пакета Intel Distribution of OpenVINO для Linux. По умолчанию файл сохраняется как `l_openvino_toolkit_p_<version>.tgz`.

Распакуйте файл `.tgz`

```
tar -xvzf l_openvino_toolkit_p_ <версия> .tgz
```

Перейдите в `l_openvino_toolkit_p_<версия>` каталог:

```
cd l_openvino_toolkit_p_ <версия>
```

Если установлена предыдущая версия инструментария Intel Distribution of OpenVINO, необходимо переименовать или удалить два каталога:

```
~/inference_engine_samples_build
```

```
~/openvino_models
```

Выберите вариант установки и следуйте инструкциям на экране.

Вариант 1: Мастер установки графического интерфейса: `sudo ./install_GUI.sh`

Вариант 2: Командная строка для инструкции: `sudo ./install.sh`

OpenVINO Tools для Linux

2. Установка зависимостей

Зависимости включают в себя:

- оптимизированную для Intel сборку библиотеки OpenCV;
- движок глубокого обучения;
- инструменты оптимизатора модели глубокого обучения;

Для установки перейдите в *install_dependencies* каталог:

```
cd /opt/intel/opencvino/install_dependencies
```

Запустите скрипт для загрузки и установки внешних программных зависимостей:

```
sudo -E ./install_opencvino_dependencies.sh
```


OpenVINO Tools для Linux

3. Задать переменные среды

Необходимо установить/обновить несколько переменных среды перед компиляцией и запуском приложения OpenVINO.

Для этого запустите следующий скрипт, чтобы временно установить переменные среды:

```
source /opt/intel/opencvino/bin/setupvars.sh
```

Необязательно: переменные среды OpenVINO удаляются при закрытии оболочки. Можно навсегда установить переменные среды следующим образом.

Открыть *bashrc* файл в директории пользователя

Редактировать файл:

```
vi <каталог_пользователя> /.bashrc
```

Добавить следующую строку в конец файла:

```
source /opt/intel/opencvino/bin/setupvars.sh
```

Сохранить и закрыть файл, для этого нажать клавишу Esc и ввести *wq*.

Для проверки необходимо открыть новый терминал. В терминале должно быть следующее сообщение:

```
[setupvars.sh] OpenVINO environment initialized.
```


OpenVINO Tools для Linux

4. Настройка оптимизатора модели

Необходимо настроить все поддерживаемые фреймворки сразу или настроить единую структуру.

Вариант 1. Настройка всех поддерживаемых платформ одновременно

Перейти в каталог обязательных компонентов оптимизатора модели:

```
cd /opt/intel/opencvino/deploy_tools/model_optimizer/install_prerequisites
```

Запустить сценарий, чтобы настроить оптимизатор моделей:

```
sudo ./install_prerequisites.sh
```

Вариант 2. Настройка каждого фреймворка отдельно

Перейти в каталог обязательных компонентов оптимизатора модели:

```
cd /opt/intel/opencvino/deploy_tools/model_optimizer/install_prerequisites
```

Запустить скрипт для необходимой модели фреймворка. Можно запустить более одного скрипта:

Для Caffe: *sudo ./install_prerequisites_caffe.sh*

Для ONNX: *sudo ./install_prerequisites_onnx.sh*

Для TensorFlow: *sudo ./install_prerequisites_tf.sh*

Для Kaldi: *sudo ./install_prerequisites_kaldi.sh*

Для MXNet: *sudo ./install_prerequisites_mxnet.sh*

OpenVINO Tools для Linux

5. Тестирование инструмента

Чтобы проверить установку и скомпилировать два примера, запустите приложения проверки. Для этого необходимо:

Перейти в демонстрационный каталог Inference Engine :

```
cd /opt/intel/opencvino/deploy_tools/demo
```

Запустить сценарий проверки классификации изображений:

```
./demo_squeezenet_download_convert_run.sh
```

Запустить сценарий проверки обнаружения транспорта:

```
./demo_security_barrier_camera.sh
```

Обзор предварительно обученных моделей OpenVINO™ Toolkit:

```
https://docs.openvino toolkit.org/latest/_models_intel_index.html
```


OpenVINO Tools для Linux

6. Оптимизирование модели в файлы IR

Перейдите в каталог:

```
cd <INSTALL_DIR>/deployment_tools/model_optimizer
```

и запустите скрипт mo.py

```
python3 mo.py --input_model INPUT_MODEL
```

Скрипт является универсальным для стандартных расширений файла модели.

Если файлы модели не имеют стандартных расширений, можно запустить скрипт с аргументом

```
--framework
```

с явным указанием типа платформы {tf,caffe,kaldi,onnx,mxnet}

Например:

```
python3 mo.py --framework tf --input_model /user/models/model.pb
```

Примечание : Некоторые модели требуют использования дополнительных аргументов , таких как

```
--scale, --scale_values, --mean_values, --mean_file.
```

Подробное описание можно найти на сайте:

https://docs.openvino toolkit.org/latest/_docs_MO_DG_prepare_model_convert_model_Converting_Model.html